2018

# Pixellated circle

Colin J. Sheppard
*Istituto Italiano di Tecnologia, University of Wollongong*

## Publication Details

Sheppard, C. J. R. (2018). Pixellated circle. Applied Optics, 57 (27), 7878-7882.

# Pixellated circle

**Abstract**
For applications in optical systems it is often necessary to represent a circular aperture in a pixellated form. An objective parameter is introduced that is a measure of how well an approximate circle can be generated from a small array of square pixels. Both filled circles (disks) and rings are considered. Arrays with a width given by an even number of pixels can also be used to generate quadrants of a circle. Rings with outer and inner profiles given by optimum circles or quadrants can be summed to fill a complete circle or quadrant.

**Publication Details**
Sheppard, C. J. R. (2018). Pixellated circle. Applied Optics, 57 (27), 7878-7882.

# Pixellated circle

**Colin J. R. Sheppard**[1,2,*]

[1] Nanophysics, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genova, Italy
[2] School of Chemistry, University of Wollongong, Wollongong NSW 2522, Australia
[*] Corresponding author: colinjrsheppard@gmail.com

*Compiled September 3, 2018*

**For applications in optical systems it is often necessary to represent a circular aperture in a pixellated form. An objective parameter is introduced that is a measure of how well an approximate circle can be generated from a small array of square pixels. Both filled circles (disks) and rings are considered. Arrays with a width given by an even number of pixels can also be used to generate quadrants of a circle. Rings with outer and inner profiles given by optimum circles or quadrants can be summed to fill a complete circle or quadrant.** 

***OCIS codes:*** (000.4430) Numerical approximation and analysis; (040.1240) Arrays; (140.3280) Laser arrays; (230.3720) Liquid crystal devices; (050.1380) Binary optics.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

## 1. INTRODUCTION

Optical systems often have circular symmetry, and nowadays pixellated devices such as detector arrays, liquid crystal devices or light emitting diode arrays are also employed. Sometimes we need to approximate a circular aperture by the pixellated structure, for example to generate a circular confocal pinhole. We are unaware of previous papers on this topic, and although several pages appear on the internet [1–4], they seem to be more concerned with the aesthetics of the approximate circle, rather than analyzing their properties. Some sites give computer code for generating pixellated circles, but they do not consider the accuracy of the results.

Representation of circular structures using a Cartesian grid also arises in sampling probles including numerical integration, convolution filters for image processing, and in finite difference time-domain (FDTD) simulations [5].But these areas are not really that similar to the present study. In computational problems, the sampling can be defined, whereas with a physical device one does not have this possibility. In the present paper the pixels are assumed to be equally weighted, which is another difference from computational work.

Here we report on a study of approximating circles by small Cartesian arrays of square pixels. We discuss different approaches to quantifying the quality of the approximate circle, and find which designs best meet our preferred criterion. We are concerned first, in Section 2, with filled circles, or disks, rather than just the circular edge, but an annular shape can be generated by subtracting one pixellated array from another, and is discussed in Section 3. The optimum arrays are found to differ in many cases from those selected by aesthetic considerations or generated from computer codes.

## 2. FILLED CIRCLES

We consider symmetrical groups of square pixels that have a width (and also height) given by an odd or even number $n$ of pixels. The structures are illustrated in Fig.1. These arrays all display four-fold symmetry (also about the 45° directions), which is important for generation of quadrants [6], as in implemention of differential phase contrast using quadrant detectors or sources, for example [7, 8]. The outside shapes vary from a square to a pixellated diamond (lozenge), with an increasing number of intermediate possible shapes as the size of the array increases. The approximation to a circle generally improves as the pixel width $n$ increases. For large arrays, the choice of particular arrangement is not so critical, so in this paper we consider only small arrays, up to $n = 16$. There are various different criteria that could be used to compare the fidelity of the pixellated approximation to a circle. Measures based on the perimeter of the array are not useful, as the perimeter $p = 4n$ for a given width is independent of its shape. Next, and perhaps simplest, is to compare the width of the structure with its area. For a perfect circle, the width is $2r$ and its area $A = \pi r^2$. So, taking each pixel as $1 \times 1$ units, the parameter

$$P_{\text{width}} = \frac{4N}{\pi n^2}, \tag{1}$$

where $N$ is the number of pixels, is unity. For a square, we have $P_{\text{width}} = 4/\pi = 1.273$. The values of this parameter for different structures are shown in Tables 1 and 2, for widths of odd and even numbers of pixels, respectively. However, this parameter does not take full account of the shape of the array. So although it gives a simple method to predict a possible design, it can also be close to unity for a bad design. Interestingly, as, in polar coordinates $r, \theta$, $p = \int r\, d\theta$, the mean radius $\bar{r} = p/2\pi = 2n/\pi$. As the area is given by $A = \frac{1}{2}\int r^2\, d\theta$, the mean square radius is

$$\langle r^2 \rangle = (N/\pi). \tag{2}$$

Then $P_{\text{width}} = (16/\pi^2)\left[\langle r^2\rangle/\bar{r}^2\right]$.

An alternative measure is based on the polar second moment of the array. The polar second moment is $I = \frac{1}{4}\int r^4\, d\theta$, giving $\langle r^4 \rangle = 2I/\pi$. The root-mean-square of the square of the radius is thus

$$(r^2)_{RMS} = \left(\frac{2\pi I - N^2}{\pi^2}\right)^{1/2}, \tag{3}$$

so that a dimensionless parameter can be defined as

$$P_{RMS} = \frac{(r^2)_{RMS}}{\langle r^2 \rangle} = \left(\frac{2\pi I}{N^2} - 1\right)^{1/2}. \tag{4}$$

This is zero for an exact circle, and $P_{RMS} = (\pi/3 - 1)^{1/2} = 0.217$ for a square. We choose this parameter, which compares $I$ and $N$, i.e. $A$, rather than one which compares $I$ with $n$, i.e. $p$, as the perimeter does not alter with $N$. For an optimum array for large $n$, $I = N^2/2\pi = \pi n^4/32$. For a particular value of $n$, the minimum value of $P_{RMS}$ tends to decrease as $n$ increases, varying as $N^{-1/2}$, or as $1/n$. Fig. 2 shows a bar chart of $N^{1/2}$ times this minimum value of $P_{RMS}$. From Fig. 2 we see that good small values of $n$ are 5, 6, 8, 9, 12 and 16. Poorer values are 4, 7, 10, 11 and 14. The design for minimum $P_{RMS}$ for any value of $n$ results in $P_{\text{width}} > 1$, except for the case $n = 4$, where two designs give the same value for $P_{RMS}$, with one ($N = 12$) giving $P_{\text{width}} < 1$. In fact, $n = 1, 2, 3$ or 4 all give the same optimum value for $P_{RMS}$. For $n = 4, N = 12$, the value of $P_{RMS}$ is the same as for a square array. The optimum value of $P_{RMS}$ also does not always coincide with the value of $P_{\text{width}}$ that is closest to unity. In Fig.1, we have shown all the possibilities for $n \leq 7$. For larger values of $n$ we show only representative cases. The array that gives the optimum value of $P_{RMS}$ is indicated by $N$ in bold type. Similarly, in Tables 1 and 2 we give values of the parameters only for some representative cases. For $n = 13$, there are two possible arrays with $N = 137$. One of these is the optimum case, but $N = 145$ is a close competitor. For $n = 14, N = 148$ and $N = 156$ give very close to the same value for $P_{RMS}$.

An interesting comparison is between the two cases, $n = 8, N = 52$ and $n = 16, N = 208$. The latter is another example where there is more than one possible arrangement. One possible shape would be a scaled version of the $n = 8$ case, so $P_{RMS}$ is also the same, but a better arrangement gives a value for $P_{RMS}$ close to one half that for $n = 8$. For values of $n$ between 6 and 12 the shape of the optimum array is of an octogonal overall form. This is because the pixels can form horizontal, vertical or 45° boundaries. But for $n \geq 13$ the shape is seen to be becoming more like a circle to the eye, as it is possible to reproduce an edge with different slopes.

## 3. PIXELLATED RINGS

A ring, or annulus, can be generated by subtraction of a smaller circle from a larger one. If the value of $n$ for the outer circle is 2 more than that for the inner circle, the ring is of thickness just a single pixel. In this case, adjacent pixels may have a common side, or their vertices may just touch. If the rings are formed by subtracting the optimum circles as described in Section 2, both

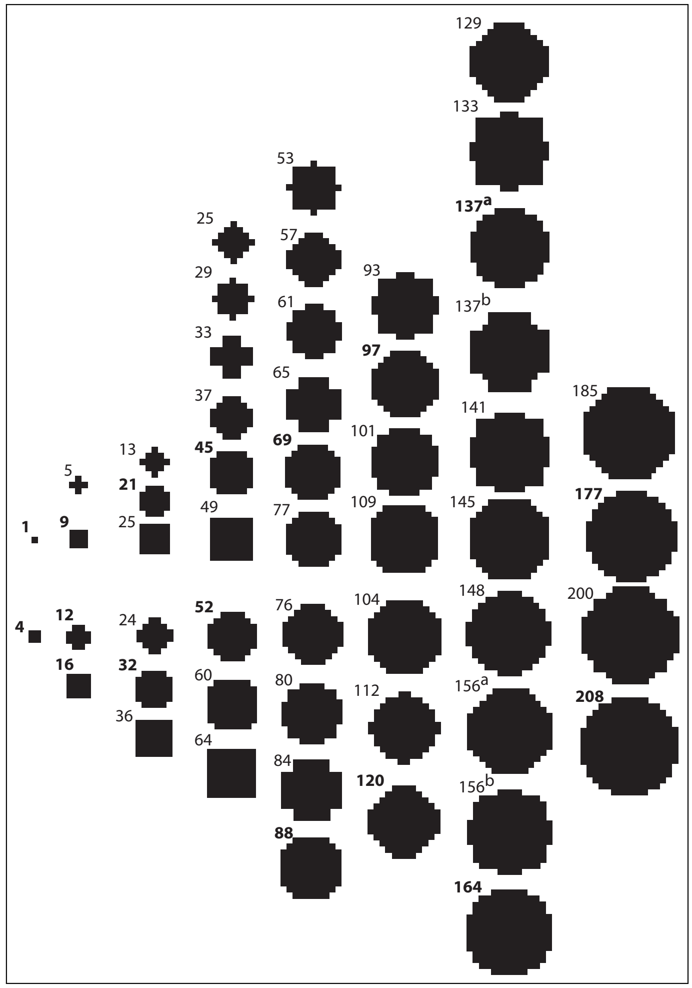


**Fig. 1.** Pixellated approximations to a circle. The number of pixels $N$ is given. The best approximations based on the parameter $P_{RMS}$ for a given $n$ are shown in bold type.

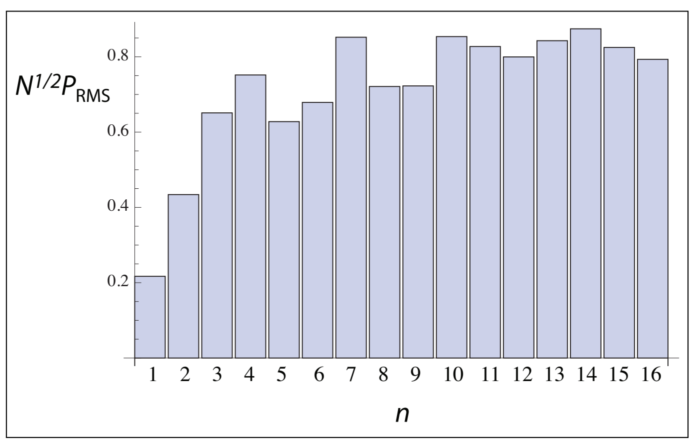


**Fig. 2.** A bar chart of $nP_{RMS}$ as a function of $n$ , for the optimum arrays for different values of $n$.

**Table 1.** The parameters $P_{\mathrm{width}}$ and $P_{RMS}$ for different arrays, for a width of an odd number of pixels. Arrays for optimum $P_{RMS}$ are shown in bold type.

| $n$ | $N$ | $P_{\mathrm{width}}$ | $P_{RMS}$ |
|---|---|---|---|
| **1** | **1** | **1.273** | **0.217** |
| 3 | 5 | 0.707 | 0.463 |
| **3** | **9** | **1.273** | **0.217** |
| 5 | 13 | 0.662 | 0.349 |
| **5** | **21** | **1.070** | **0.137** |
| 5 | 25 | 1.273 | 0.217 |
| 7 | 25 | 0.650 | 0.296 |
| 7 | 29 | 0.754 | 0.228 |
| 7 | 33 | 0.857 | 0.306 |
| 7 | 37 | 0.961 | 0.140 |
| **7** | **45** | **1.169** | **0.127** |
| 7 | 49 | 1.273 | 0.217 |
| 9 | 41 | 0.644 | 0.528 |
| 9 | 53 | 0.833 | 0.199 |
| 9 | 57 | 0.896 | 0.155 |
| 9 | 61 | 0.959 | 0.130 |
| 9 | 65 | 1.022 | 0.165 |
| **9** | **69** | **1.085** | **0.087** |
| 9 | 77 | 1.210 | 0.150 |
| 9 | 81 | 1.273 | 0.217 |
| 11 | 89 | 0.937 | 0.464 |
| 11 | 93 | 0.979 | 0.140 |
| **11** | **97** | **1.021** | **0.084** |
| 11 | 101 | 1.063 | 0.090 |
| 11 | 109 | 1.147 | 0.101 |
| 11 | 117 | 1.231 | 0.167 |
| 11 | 121 | 1.273 | 0.217 |
| 13 | 129 | 0.971 | 0.097 |
| 13 | 133 | 1.002 | 0.279 |
| **13** | **137**[a] | **1.032** | **0.072** |
| 13 | 137[b] | 1.032 | 0.126 |
| 13 | 141 | 1.062 | 0.112 |
| 13 | 145 | 1.092 | 0.073 |
| **15** | **177** | **1.002** | **0.062** |
| 15 | 185 | 1.047 | 0.066 |

**Table 2.** The parameters $P_{\mathrm{width}}$ and $P_{RMS}$ for different arrays, for a width of an even number of pixels. Arrays for optimum $P_{RMS}$ are shown in bold type.

| $n$ | $N$ | $P_{\mathrm{width}}$ | $P_{RMS}$ |
|---|---|---|---|
| **2** | **4** | **1.273** | **0.217** |
| **4** | **12** | **0.955** | **0.217** |
| **4** | **16** | **1.273** | **0.217** |
| 6 | 24 | 0.849 | 0.217 |
| **6** | **32** | **1.132** | **0.120** |
| 6 | 36 | 1.273 | 0.217 |
| 8 | 44 | 0.875 | 0.173 |
| **8** | **52** | **1.035** | **0.100** |
| 8 | 60 | 1.194 | 0.139 |
| 8 | 64 | 1.273 | 0.217 |
| 10 | 68 | 0.866 | 0.135 |
| 10 | 72 | 0.917 | 0.167 |
| 10 | 76 | 0.968 | 0.111 |
| 10 | 80 | 1.019 | 0.103 |
| 10 | 84 | 1.070 | 0.137 |
| **10** | **88** | **1.120** | **0.091** |
| 10 | 100 | 1.273 | 0.217 |
| 12 | 104 | 0.920 | 0.127 |
| 12 | 112 | 0.990 | 0.084 |
| 12 | 116 | 1.026 | 0.122 |
| **12** | **120** | **1.061** | **0.073** |
| 12 | 124 | 1.096 | 0.089 |
| 14 | 148 | 0.961 | 0.07675 |
| 14 | 156 | 1.013 | 0.07673 |
| **14** | **164** | **1.065** | **0.070** |
| 14 | 172 | 1.117 | 0.080 |
| 16 | 200 | 0.995 | 0.078 |
| **16** | **208** | **1.035** | **0.055** |

**Table 3.** Properties of the optimum pixellated rings.

| $n$ | $N$ | $3I$ | $(I/N)^{1/2}$ | $r_2 - r_1$ |
|---|---|---|---|---|
| 3 | 8 | 40 | 1.291 | 1.087 |
| $4^a$ | 8 | 64 | 1.633 | 0.804 |
| $4^b$ | 12 | 120 | 1.826 | 1.097 |
| 5 | 12 | 174 | 2.198 | 1.710 |
| $6^a$ | 20 | 424 | 2.658 | 1.231 |
| $6^b$ | 16 | 368 | 2.769 | 0.933 |
| 7 | 24 | 768 | 3.266 | 1.189 |
| 8 | 20 | 808 | 3.670 | 0.874 |
| 9 | 24 | 1308 | 4.262 | 0.901 |
| 10 | 36 | 2424 | 4.738 | 1.220 |
| 11 | 28 | 2236 | 5.159 | 0.867 |
| 12 | 32 | 3184 | 5.759 | 0.887 |
| 13 | 40 | 4482 | 6.112 | 1.046 |
| 14 | 44 | 5992 | 6.738 | 1.043 |
| 15 | 40 | 6008 | 7.076 | 0.902 |
| 16 | 44 | 7816 | 7.694 | 0.912 |

the outer and inner boundaries are optimum. This also has the property that the rings can be summed to completely cover a filled circle. The analogy of our measure $P_{\text{width}}$ is

$$P_{\text{width}} = \frac{N}{\pi(n-1)}, \tag{5}$$

where the factor $n-1$ in the denominator comes from the average between the outer and inner widths. For a square, $N = 4(n-1)$, so $P_{\text{width}} = 4/\pi = 1.273$ as before. For a diamond shape, $N = 2(n-1)$, so $P_{\text{width}} = 2/\pi = 0.637$. The optimum pixellated rings are shown in Fig.3. Two alternative arrangements are given (labelled $a$ and $b$) for $n = 4, 6$, as there were two equally good arrangements for a $4 \times 4$ or $6 \times 6$ circle.

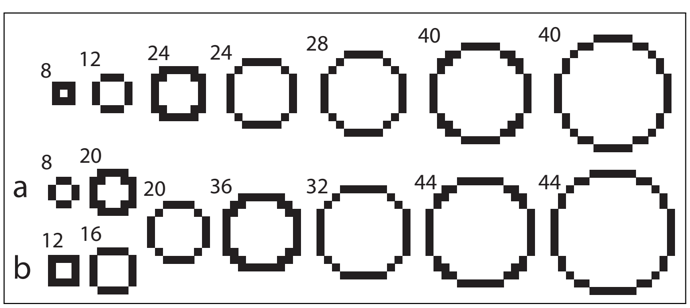


**Fig. 3.** The optimum pixellated rings of thickness one pixel, for odd $n$ from 3 to 15 (upper), and even $n$ from 4 to 16 (lower). The numbers labelled are the corresponding values of $N$ . There are two different equally good solutions for $n = 4$ and $n = 6$ according to our criterion (labelled $a$ and $b$).

For particular values of $N, I$, the array is an approximation to

**Table 4.** Preferred values of $N$ according to different sources. Values the same as our optimum are shown in bold type.

| $n$ | $N$ Imgur, outside [3] | $N$ Imgur, inside [3] | $N$ Minecraft, outside [4] | $N$ Minecraft, inside [4] | $N$ Barrett [2] |
|---|---|---|---|---|---|
| 1 | **1** | **1** | - | - | - |
| 2 | **4** | **4** | - | - | **4** |
| 3 | **9** | **9** | - | **9** | **9** |
| 4 | **12** | **16** | - | **12** | **12** |
| 5 | **21** | **21** | **21** | **21** | **21** |
| 6 | **32** | **32** | 24 | 24 | **32** |
| 7 | 37 | **45** | 37 | 37 | 37 |
| 8 | **52** | **52** | 44 | 44 | 44, **52** |
| 9 | **69** | **69** | 61 | 61 | 61, **69** |
| 10 | 80 | 80 | 68 | 72 | 68, **88** |
| 11 | **97** | 101 | 89 | 93 | 89, 101 |
| 12 | 112 | **120** | 104 | *120* | 96, 112 |
| 13 | **137** | **137** | 125 | **137** | 121, **137** |
| 14 | **156** | 164 | *156* | *156* | 148, 164 |
| 15 | **177** | **177** | **177** | **177** | 157, **177** |
| 16 | **208** | **208** | 200 | 200 | 188, **208** |

**Table 5.** For pixellated rings, the values of N from other sources compared with our optimum values

| $n$ | $N$, optimum | $N$, Imgur [3] | $N$, Minecraft [4] |
|---|---|---|---|
| 3 | 8 | **8** | - |
| 4 | 8 | **8** | - |
| 5 | 12 | **12** | **12** |
| 6 | 20 | 16 | 12 |
| 7 | 24 | 16 | 16 |
| 8 | 20 | **20** | *20* |
| 9 | 24 | **24** | *24* |
| 10 | 36 | 28 | 24 |
| 11 | 28 | **28** | *28* |
| 12 | 32 | 32 | 40 |
| 13 | 40 | 36 | 32 |
| 14 | 44 | 36 | 36 |
| 15 | 40 | **40** | **40** |
| 16 | 44 | **44** | *44* |

a true circular annulus with outer and inner radii $r_2, r_1$, respectively. The areal mean radius (AMR) is given by

$$AMR = \left[\frac{1}{2}(r_1^2 + r_2^2)\right]^{1/2} = \left(\frac{I}{N}\right)^{1/2} \qquad \textbf{(6)}$$

The effective thickness of the annulus is

$$r_2 - r_1 = \left(\frac{N}{2\pi}\right)^{1/2}\left[\left(\frac{2\pi I}{N^2}+1\right)^{1/2} - \left(\frac{2\pi I}{N^2}-1\right)^{1/2}\right] \qquad \textbf{(7)}$$

The properties of the optimum pixellated rings are given in Table 3. We find that the AMR increases close to linearly with $n$, and for large $n$ tends to a value $(n-1)/2$. The effective thickness tends to unity, so the fractional thickness becomes smaller for larger $n$.

As the rings of thickness one pixel are contiguous, rings of different integer values of thickness can be generated by addition.

## 4. DISCUSSION

An objective parameter has been introduced that measures the accuracy of a pixellated approximation to a circle. This parameter is based on the second moment of area of the pixellated array. It determines the optimum number and arrangement of the pixels for a given array width. Preferred arrays, given on the web, predicted by aesthetic considerations or generated by computer codes, are shown in Table 4 [2–4]. Some of these websites show suggestions for pixellated circular rings, giving circles defined by either the outer or the inner boundaries. Values for the number of pixels $N$ equal to our optimum values are indicated in bold type. For some cases, the value of $N$ is the same as ours, but the pixel arrangements (and also $P_{RMS}$) are different. These are indicated by italic type. Minecraft's arrays for $N = 13$ and $N = 14$ do not have four-fold symmetry. It is seen that there are substantial differences in the preferred arrangements for $n > 5$, indicating that generation of a pixellated circle is by no means an obvious process. We have not found a simple formula for generating optimum arrays. This seems difficult as a result of discretization. We have identified some general characteristics: in all cases given, the length of the level parts (say horizontal) of the perimeter never increase with distance from the center line. However, the length of the longest level part may decrease with increasing $n$, e.g. for $n = 10, 12$.

Pixellated rings have been constructed by subtracting two optimum pixellated circles. These have optimum outer and inner boundaries. They also sum to fill the interior of a pixellated circle, unlike pixellated rings in the literature. For pixellated rings, the values of $N$ from other sources are compared with our optimum values in Table 5. Imgur's preferred arrays coincide with ours for only half of the cases. Minecraft's preferred arrays are the same as our optimum only in one case. For another three cases, the value of $N$ is the same as ours, but the pixel arrangements (and also $P_{RMS}$) are different, indicated by italic type. Minecraft's arrays for $N = 13$ and $N = 14$ do not even have four-fold symmetry.

Finally, the pixellated rings with even pixel width described here are particularly suited to generating quadrant ring arrays. For example a array of sources or detectors can be used to generate eight quadrant rings. Such an array should be particularly useful for applications in computational imaging [9, 10].

## REFERENCES


1. Wikipedia, https://en.wikipedia.org/wiki/Midpoint circle algorithm.
2. S. Barrett, http://silverspaceship.com/articles/pixelated circles.html (2011).
3. Imgur, http://i.imgur.com/SPBdV.png.
4. Minecraft, http://www.minecraftguides.org/circles/.
5. A. Farjadpour, D. Roundy, A. Rodriguez, M. Ibanescu, P. Bernel, J. D. Joannopoulos, and S. G. Johnson, "Improving accuracy by subpixel smoothing in the finite-difference time domain," Opt. Lett. **31**, 2972–2974 (2006).
6. M. Castello, C. J. R. Sheppard, A. Diaspro, and G. Vicidomini, "Image scanning microscopy with a quadrant detector," Opt. Lett. **40**, 5355–5358 (2015).
7. D. K. Hamilton and C. J. R. Sheppard, "Differential phase contrast in scanning optical microscopy," J. Microsc. **133**, 27–39 (1984).
8. S. B. Mehta and C. J. R. Sheppard, "Quantitative phase-gradient imaging at high resolution with asymmetric illumination-based differential phase contrast," Opt. Lett. **34**, 1924–1926 (2009).
9. D. K. Hamilton, C. J. R. Sheppard, and T. Wilson, "Improved imaging of phase gradients in scanning optical microscopy," J. Microsc. **153**, 275–286 (1984).
10. L. Tian and L. Waller, "3D intensity and phase imaging from light field measurements in an LED microscope," Optica. **2**, 104–111 (2015).